Principal Component Analysis (PCA): Mathematical Foundations and Intuitive Explanation

Ahmed BADI
ahmedbadi905@gmail.com
linkedin.com/in/badi-ahmed

January 2026

Abstract

Principal Component Analysis (PCA) is one of the most widely used dimensionality reduction techniques in machine learning and data analysis. It transforms a set of possibly correlated features into a smaller set of uncorrelated components that capture most of the variability in the data. This article explains the intuition behind PCA, derives its mathematical formulation using covariance matrices and eigendecomposition, and shows how to use it for dimensionality reduction and data visualization. We also discuss practical aspects such as data standardization, choosing the number of components, and limitations of PCA as a linear method.

Keywords: Principal Component Analysis, Dimensionality Reduction, Eigendecomposition, Covariance Matrix, Variance, Feature Extraction.

1 Introduction

High-dimensional datasets are common in modern applications: images, sensor data, text, genomics and more. Working directly with many features can be difficult, leading to high computational cost and overfitting. PCA provides a simple and powerful way to reduce dimensionality while preserving as much information (variance) as possible. [1], [2]

PCA finds a new coordinate system where:

- The axes (principal components) are linear combinations of the original features.
- The components are orthogonal (uncorrelated).
- The first component captures the largest possible variance, the second the next largest, and so on.

By keeping only the first few components, we obtain a low-dimensional representation that often simplifies modeling and visualization. [3], [4]

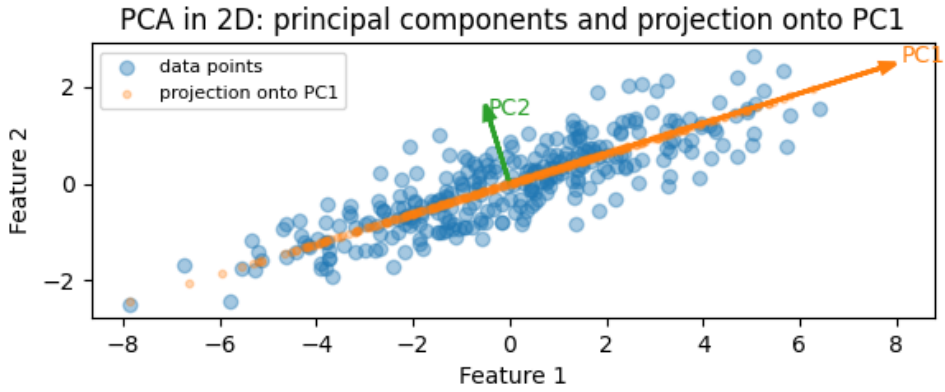

Figure 1: PCA finds a new axis (PC1) that captures the maximum variance in the data.

2 Goals of PCA

The main goals of PCA can be summarized as: [2], [5]

- **Dimensionality reduction:** Represent data with fewer variables while retaining most of the variation.
- **De-correlation:** Transform correlated features into uncorrelated components.
- **Feature extraction:** Create new features (principal components) that may be more informative for downstream tasks.
- **Visualization:** Project high-dimensional data to 2D or 3D for plotting.

Mathematically, PCA seeks a linear transformation that maximizes variance along each new axis, under orthogonality constraints.

3 Mathematical Formulation

3.1 Data Matrix and Centering

Consider a dataset with n samples and p features. Let $X \in \mathbb{R}^{n \times p}$ denote the data matrix, where each row is a sample $\mathbf{x}_i^\top$. Before applying PCA, we typically center each feature to have mean zero:

$$\tilde{\mathbf{x}}_i = \mathbf{x}_i - \bar{\mathbf{x}},$$

where

$$\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i.$$

Let $\tilde{X}$ be the centered data matrix.

3.2 Covariance Matrix

The sample covariance matrix $\Sigma \in \mathbb{R}^{p \times p}$ is defined as [1], [3]

$$\Sigma = \frac{1}{n-1} \tilde{X}^\top \tilde{X}.$$

Its entries are

$$\Sigma_{jk} = \frac{1}{n-1} \sum_{i=1}^{n} (\tilde{x}_{ij})(\tilde{x}_{ik}),$$

which measure how features j and k vary together.

3.3 Eigendecomposition of the Covariance Matrix

PCA is based on the eigendecomposition of Σ:

$$\Sigma \mathbf{w}_k = \lambda_k \mathbf{w}_k,$$

where:

- λ_k are eigenvalues ($\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$);
- $\mathbf{w}_k$ are corresponding eigenvectors (principal directions), which can be chosen orthonormal.

Each eigenvalue λ_k represents the variance captured by the k-th principal component, and each eigenvector $\mathbf{w}_k$ defines the direction of that component in feature space. [2], [6]

3.4 Principal Components

The k-th principal component scores are obtained by projecting the data onto $\mathbf{w}_k$:

$$z_{ik} = \mathbf{w}_k^\top \tilde{\mathbf{x}}_i,$$

or in matrix form, for the first m components:

$$Z = \tilde{X} W_m,$$

where $W_m = [\mathbf{w}_1, \ldots, \mathbf{w}_m] \in \mathbb{R}^{p \times m}$ and $Z \in \mathbb{R}^{n \times m}$ is the transformed data in the new coordinate system. [3]

3.5 Variance Maximization View

The first principal component $\mathbf{w}_1$ solves:

$$\max_{\mathbf{w}} \ \mathbf{w}^\top \Sigma \mathbf{w} \quad \text{s.t. } \|\mathbf{w}\|_2 = 1.$$

This optimization has the solution $\mathbf{w} = \mathbf{w}_1$, the eigenvector with the largest eigenvalue λ_1. Similarly, the k-th principal component is the eigenvector with the k-th largest eigenvalue, subject to orthogonality to previous components. [7], [8]

4 Dimensionality Reduction with PCA

4.1 Explained Variance

The total variance in the data is the sum of all eigenvalues:

$$\text{TotalVariance} = \sum_{j=1}^{p} \lambda_j.$$

The variance explained by the k-th principal component is λ_k, and the proportion of variance explained (PVE) by the first m components is

$$\text{PVE}(m) = \frac{\sum_{k=1}^{m} \lambda_k}{\sum_{j=1}^{p} \lambda_j}.$$

In practice, we choose the number of components m such that $\text{PVE}(m)$ exceeds a threshold, such as 90% or 95%. [1], [9]

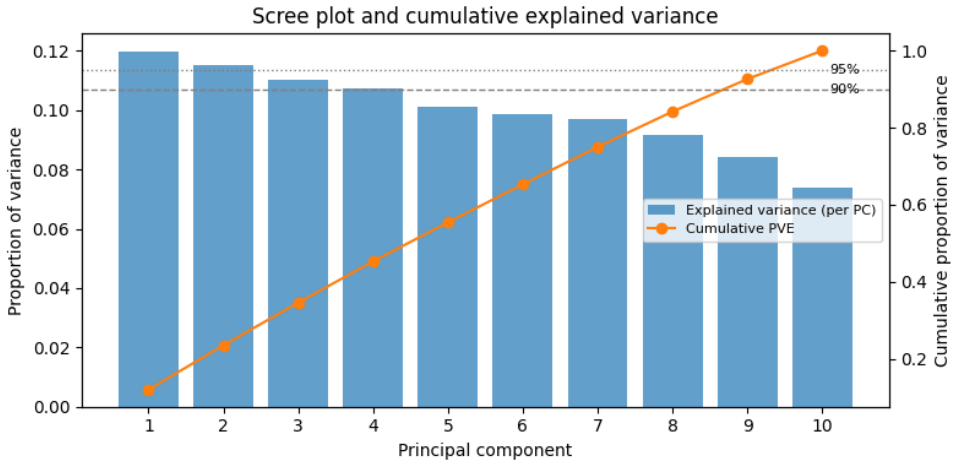

Figure 2: Scree plot and cumulative explained variance used to choose the number of principal components.

4.2 Projection and Reconstruction

After choosing m, we project centered data onto the first m principal components:

$$Z = \tilde{X} W_m.$$

To approximately reconstruct the original data from the reduced representation:

$$\hat{X} = Z W_m^\top + \mathbf{1}\bar{\mathbf{x}}^\top,$$

where $\mathbf{1}$ is an n-dimensional column vector of ones. The reconstruction error is

$$\|\tilde{X} - \hat{X} + \mathbf{1}\bar{\mathbf{x}}^\top\|_F^2,$$

which equals the sum of the discarded eigenvalues times $(n-1)$. [10]

5 PCA via Singular Value Decomposition (SVD)

An alternative and numerically stable way to compute PCA is via the Singular Value Decomposition of the centered data matrix. [2], [7]

Let

$$\tilde{X} = U S V^\top,$$

where:

- $U \in \mathbb{R}^{n \times p}$ has orthonormal columns (left singular vectors),
- $S \in \mathbb{R}^{p \times p}$ is diagonal with singular values $s_1 \geq s_2 \geq \cdots \geq s_p \geq 0$,
- $V \in \mathbb{R}^{p \times p}$ has orthonormal columns (right singular vectors).

Then:

$$\Sigma = \frac{1}{n-1} \tilde{X}^\top \tilde{X} = \frac{1}{n-1} V S^2 V^\top.$$

Thus:

- Eigenvectors of Σ are columns of V.
- Eigenvalues are $\lambda_k = \frac{s_k^2}{n-1}$.

Principal components can be computed directly as

$$Z = \tilde{X}V_m = U_m S_m,$$

where V_m contains the first m columns of V, and U_m, S_m are the corresponding truncated matrices. [3], [4]

6 Geometric Interpretation

Geometrically, PCA performs a rotation (and possibly reflection) of the coordinate axes so that:

- The first axis points in the direction of maximum variance.
- Each subsequent axis is orthogonal to the previous ones and captures remaining variance.

In 2D, this corresponds to finding a line that best fits the data in a least-squares sense, then an orthogonal line capturing the remaining spread. In higher dimensions, PCA finds a best-fit subspace. [7], [9]

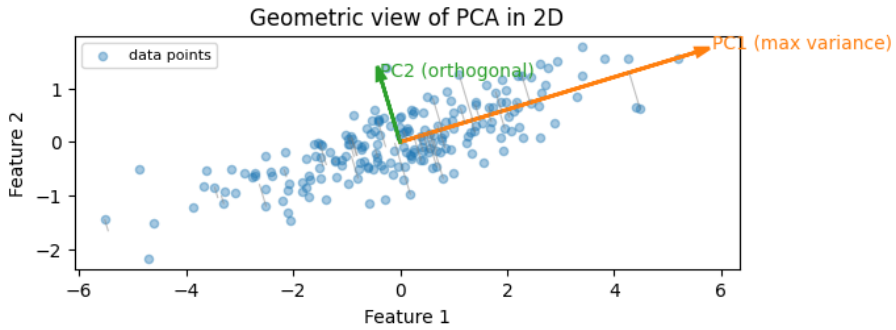

Figure 3: Geometric view of PCA in 2D: PC1 aligns with the direction of maximum variance.

7 Practical Considerations

7.1 Standardization

When features are on different scales (for example, height in centimeters, weight in kilograms), PCA is sensitive to feature scaling. Features with larger variance dominate the first components. [1], [11]

Common practice:

- Standardize each feature to zero mean and unit variance before PCA:

$$x'_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j},$$

where s_j is the standard deviation of feature j.

7.2 Choosing the Number of Components

Methods to choose m: [9], [12]

- **Explained variance threshold** (e.g., 90%).
- **Scree plot:** Look for an "elbow" where adding more components yields diminishing returns.
- **Cross-validation:** Evaluate model performance with different numbers of components.

7.3 Limitations

- PCA is linear; it cannot capture nonlinear manifolds.
- Principal components are combinations of original features and may be hard to interpret.
- PCA maximizes variance, which may not always align with class separability (LDA may be better for supervised tasks). [13]

8 Applications

PCA is widely used in: [2], [3]

- **Preprocessing:** Dimensionality reduction before clustering or classification.
- **Noise reduction:** Removing low-variance components often denoises data.
- **Visualization:** Plotting high-dimensional data in 2D or 3D via first components.
- **Image compression and eigenfaces:** Representing faces as linear combinations of principal components.

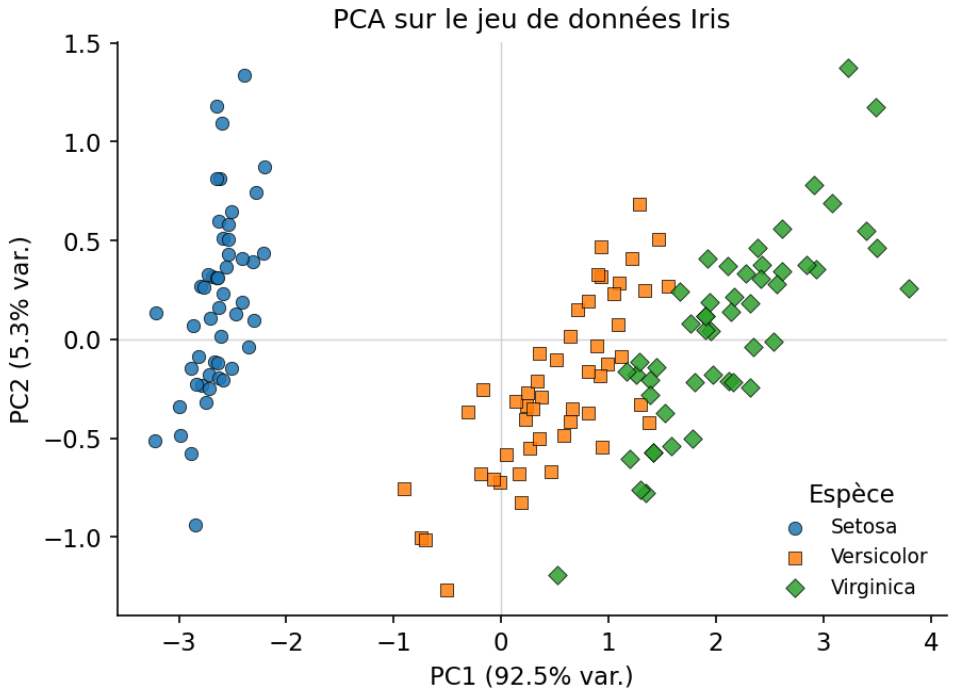

Figure 4: PCA applied to the Iris dataset: classes visualized in the space of the first two principal components.

9 Conclusion

Principal Component Analysis is a fundamental tool for dimensionality reduction and feature extraction. Its main strengths are:

- Simplicity and broad applicability.
- Clear mathematical foundation via covariance matrices and eigendecomposition.

- Ability to decorrelate features and concentrate information into a few components.

At the same time, PCA is limited by its linear nature and the sometimes opaque interpretation of components. For nonlinear structure, methods like kernel PCA, Isomap, LLE, t-SNE and UMAP may be more appropriate. Nevertheless, PCA often serves as a strong baseline and a first step in exploratory data analysis. [2], [14]

References

[1] GeeksforGeeks, *Mathematical approach to pca*, https://www.geeksforgeeks.org/machine-learning/mathematical-approach-to-pca/, 2021.

[2] Wikipedia contributors, *Principal component analysis*, https://en.wikipedia.org/wiki/Principal_component_analysis, 2024.

[3] S. Raschka, *Principal component analysis in 3 simple steps*, https://sebastianraschka.com/Articles/2015_pca_in_3_steps.html, 2015.

[4] B. In, *A step-by-step explanation of principal component analysis*, https://builtin.com/data-science/step-step-explanation-principal-component-analysis, 2025.

[5] GeeksforGeeks, *Mathematical approach to pca*, https://www.geeksforgeeks.org/machine-learning/mathematical-approach-to-pca/, 2021.

[6] B. Hanson, *The math behind pca*, https://bryanhanson.github.io/LearnPCA/articles/Vig_06_Math_Behind_PCA.html, 2024.

[7] J. Shlens, *A tutorial on principal component analysis*, https://www.cs.cmu.edu/~elaw/papers/pca.pdf, 2014.

[8] CodeSignal, *Eigenvectors, eigenvalues, and covariance matrix explained*, https://codesignal.com/learn/courses/navigating-data-simplification-with-pca/lessons/mastering-pca-eigenvectors-eigenvalues-and-covariance-matrix, 2024.

[9] J. Starmer, *Principal component analysis (pca), step-by-step*, https://www.youtube.com/watch?v=FgakZw6K1QQ, 2018.

[10] Neuromatch Academy, *Tutorial 3: Dimensionality reduction & reconstruction*, https://compneuro.neuromatch.io/tutorials/W1D4_DimensionalityReduction/student/W1D4_Tutorial3.html, 2019.

[11] Turing, *Step-by-step guide to principal component analysis*, https://www.turing.com/kb/guide-to-principal-component-analysis, 2025.

[12] G. Miani, *Principal component analysis (pca): Explained step-by-step*, https://builtin.com/data-science/step-step-explanation-principal-component-analysis, 2025.

[13] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[14] GeeksforGeeks, *Introduction to dimensionality reduction*, https://www.geeksforgeeks.org/machine-learning/dimensionality-reduction/, 2017.